BOSTON COMPUTER SOCIETY
Newsletter of the BoSTUG (Sinclair/Timex User Group)
Vol 7 No 5

EDITOR'S NOTES

Time Designs Magazine recently had a newsletter competition for Sinclair/Timex User Groups. We submitted our May issue. Surprisingly, we did not finish last.

Nor, did we finish first.

Said the judges, ".... clean production. Interesting articles ... leaning more heavily to the QL. This Sinclair SIG has been around since "day one". It has had some ups and downs ..."

Rarely does a judgee agree with a judge, but we agree entirely. An editor with a QL knows others who sometimes let us twist their arms and our active group is heavily into QLs.

On this page we have sought contributions from other Sinclairs but, with notable exceptions, the request has fallen on deaf ears. (Past pleas for members to let us know what turns them on have been lonely cries in the wilderness, so we draw your attention to this month's centerfold.)

Not commented on by the judges, but one reason we tied for eighth in a field of 22, was a lack of any distinguished graphics.

To correct this deficiency we invite you to crank up your creative juices and submit designs for the following:

1. A Newsletter cover. (The BCS logo is available in paste art.)

2. A Banner Head (See page top).

To encourage you to squeeze the juices, a year's subscription to Time Designs (see review) will be the first prize and you will have your name immortalized in print.

DIRECTORY

Co-director	Peter Hale (617) 889-0830
Co-director	John Kemeny (508) 263-3347
General Questions	John Kemeny (508) 263-3347
Machine Language	John Kemeny (508) 263-3347
Software Librarian	Jim Rodlin (508) 481-2155
SuperBASIC Guru	Al Boehm (617) 275-1097
Editor (pro tem)	Peter Hale (617) 889-0830
QL sub-Group	Peter Hale (617) 889-0830
Flexi-BBS	(617) 648-7651
Sysop: Bob Cutter	646-4425
THE TIMEWARP(BBS	(508) 481-0555
Sysop: Jim Rodlin	481-2155
Member Services Liason	Will Stackman (617) 547-0655

The Sinclair/Timex Newsletter is the official publication of the Sinclair/Timex User Group of the Boston Computer Society (BoSTUG).

Annual membership for New England residents is $35.00 (otherwise $28.00) per year. Subscription to BCS/Update and one other newsletter is included without cost.

The newsletter has an exchange policy with Sinclair/Timex User groups willing to reciprocate. Please forward exchange issues to the editor pro tem, Peter Hale, P.O. Box 8763, Boston, MA 02114.

Unless copywritten by the author, articles are in the public domain. Attribution is requested.

CONTENTS

	page
Editor's Notes	1
Directory	1
WHEN_ERROR	2
Psion Productivity Tips	2
Calendar Notes	2
Snug as a Bug	3
Snug Conference	4
Exchange Newsletters	5
CS 2068 Machine Code-Review	5
Magazine Reviews	6
Larken Ramdisk	7
Quill Filesaver	8
BCS Programming Project	9
BINGO	9
East Coast Sync Fest	10
About Toolkit II	11
Columnifying Quill	12

WHEN_ERROR

The listing in the Atomic Powered QL, in the last issue had an error. Line 100 should read:

```
100 IF LEN(u$)<>20 THEN u$="":
    GO TO 40
```

Western papers, please copy.

PSION PRODUCTIVITY TIPS

In the three years since getting a QL, I continue to be surprised at little secrets it reveals.

Here is a new one and two oldies some haven't tumbled to yet.

1 On screen, the number zero (0) has a slash so it looks different from the capital letter O. Most printers have a dip switch setting so the zero can be printed without the slash. Pretty, but once in a while you actually want the slash in the zero, as when printing a listing from Quill.

In the bowels of the QL's character set is <CTRL><SHIFT> P which is like slash zero. QLs and many Epson/Seikoshas can print this.

2 In the Psion programs there is a screen re-draw key <SHIFT><F5>. Screen redrawing rewrites the screen. It's useful in Quill if on-screen text jumps all over the place during editing.

When inserting text in the middle of a document, space opens in the middle of the screen to let Quill give you and it some room. Add a character in the middle of text and everything after is pushed one byte in RAM. Keep this up at a rapid pace and the screen gets stressed trying to keep up. At the end, text doesn't always close up neatly. <SHIFT><F5> is Psion's equivalent of Valium.

3 Speaking of <SHIFT><F5>, not everyone knows about <CTRL><F5>, which pauses on-screen scrolling.

In SuperBASIC, without Toolkit II extensions (and sometimes with), and in programs which send all kinds of stuff to the screen but have no built-in pause, it can be desirable to pause the program.

LISTing a program, COPYing a file to scr or doing a long DIRectory, can be frustrating. Scrolling is paused by pressing <CTRL><F5>; to continue, press <CTRL><F5> again.

Actually, any key press continues scrolling but CTRL F5 is best. It does not register in the buffer and consequently is not mistaken for a subsequent INKEY$, or its equivalent, as a keyed response.

- Peter Hale

CALENDAR NOTES

The Sinclair Timex User Group General meeting will be held on October 19, 1988, at 7:30 pm at the U. Mass Harborside Campus on Columbia Point. The room is 063 Wheatly Building.

The QL sub-group next meets on December 5 at John Mitchell's in Westwood. As this is a private home, please call John at (617) 326-5420 so he can tell you how to get there.

For other information contact Peter Hale or John Kemeny.

SNUG AS A BUG

On cold evenings of my childhood my grandmother would wrap me in a handwoven blanket and settle in front of the fire to read me a story. But before the story she would always say, " and there you are, snug as a bug in a rug." The sense of being protected has lasted with me to this day every time I hear the word 'snug'.

In 1988 SNUG is the acronym for Sinclair North American Users Group. It's objective is to support all Sinclairs across this vast continent, maintain PD and shareware software libraries, act as a referal service, publish newsletters and generally promote Sinclair computers. It is to complement, not compete with, local user groups.

All worthy goals. Realistic? Who's to say? Some wet blankets have editorially mused that it is a bug in a rug, because of ill-defined organization, naive economics and avoidance of reality.

Yet, for all the talk of the Sinclair being 'dead', consider:

1. Hardware peripherals for the T/S 1000 continue to come to market and survive.

2. The Larken disk controller for the T/S 2068 is a technical and financial success.

3. Time Designs Magazine turns a profit.

4. Sinclair computers are virtually the only ones that permit novice hackers to understand the inner workings. Somewhere, there have to be training grounds to get hands-on, basic understanding of machine code.

Sinclairs are not dead nor will they die. Like old soldiers they will fade away, but in the meantime thousands of Sinclair owners power up and get productive use, vast entertainment and much satisfaction from their computers.

The arguement for SNUG is that if smaller groups disband, there can be a vehicle for continuing support and assistance to new users as they come into the fold. New Users? Yes, daily new people are turned on to Sinclairs and there is no media attention to connect these people to the experienced.

Mel Nathanson, 7515 Arbordale Dr. Port Richey, FL 34668 is collecting names and addresses of those who are interested in the concept and/or want more info. His phone is (813) 863-5552.

Drop him a postcard. At worst you're out 15¢ (17¢ if you wait)

- Peter Hale

APPLICATIONS SOFTWARE for the QL

from *EMSOFT*

MAILBAG An address database you won't believe. Labels, Rotary index cards, Zip sorting, Tickler filing, 3 column printout, many **automatic features** and much more 256K+

TAX-I-QL Spreadsheet for IRS tax returns 384K+

QLAND_LORD Manage up to 99 units and do Schedule E. (Soon) 256K+

TRUST_FUND Track investments an net worth. Retain cost basis (Soon) 256K+

$19.95 ea ppd on 5 1/4" DSQD
(+$2.00 on microcartridge)

EMSOFT
P.O. Box 8763, Boston, MA 02114
(617) 889-0830

SNUG CONFERENCE

(Minutes of the SNUG (Sinclair North American User Group) conference of 8/17/88 on Compuserve)

We met on the Timex section of Compuserve on Wednesday, August 17 at 9pm to discuss SNUG. Among the 13 active participants were: Mel Nathanson (Chairman pro-tem of SNUG), Gary Lessenberry (of CATUG), Paul Holmgren (ISTUG), Sysop Dave Rothman (Compuserve), Jim Rodlin (the Boston Computer Society), Greg Popovich (of LIST) and several others.

The meeting was active, and many interesting ideas were proposed. There were two primary topics:
A) The SNUG Newsletter, and
B) The SNUG Public Domain Library

At the beginning of the discussion, since we were all confirmed modem users, it was suggested that a BBS be the central point for SNUG. But it was acknowledged that many users either cannot or will not use their modems (for whatever reason). So even as the idea to disseminate the newsletter from a BBS was proposed, it was eliminated. It was decided that any newsletter would HAVE to be in hard-copy format. It was also felt that SNUG should definitely have its own separate newsletter rather than depend on other publication(s). This would give a stronger sense of identity on the part of the membership.

As far as the Public Domain Library went, consensus was more difficult to reach. In the beginning, once-again, it was proposed that the library should be up for downloading. But this was eliminated, largely for the same reasons as the newsletter. Then, it was proposed that we confine it strictly to tape media since everyone can read tape. But the objection was raised that this would consttute a very big demand on the time of the Librarian. It was then suggested that the disk formats should also be supported. The mechanics of such a system were discussed for some time, the most prominent idea being to have "Sub-Librarians" to handle the various specific formats. But no firm conclusion was reached.

The idea of setting up an Echomail system based on Fidonet was briefly discussed. Two T/S BBSes are currently based on Fidonet nodes with a third to be added soon. The current ones are the Passaic BBS (node 107/559) and the Bus Depot (node 112/4). There seemed to be some enthusiasm for the idea.

The last item of business was that we agreed to meet again on Compuserve on Wednesday, August 31st at 9:00pm EDT.

- Pete Fischer
("Phoenix Pete")

We found a small supply of ZX Spectums and ZX Interface 1's in their original cartons.

They won't last at these prices!

ZX Spectrum with 240v PS (or use TS 2068 PS) $40.00

ZX Interface 1 $30.00

(plus $5.00 s+h ea)

Visa/Master Card accepted

ask for Neal.

And ask about our prices on wide carriage, daisy wheel QUME serial printers

ELI HEFFRON & Sons inc.
Computer Equipment & Peripherals

139-145 Hampshire St.
Cambridge, MA 02139

1-800-892-9230 1-617-547-4005

EXCHANGE NEWSLETTERS

Since renewing our publication schedule in January, BoSTUG gets exchange newsletters from many groups across the continent.

The newsletters are available to members at the bi-monthly meetings. When someone figures out how to keep track of loan-outs, they will be available for take-out and mailout. Suggestions!

Quality and quantity vary. The most reliable is the HATS from Lemoyne, PA, with whom we tied for eighth place in the Time Designs Magazine contest in May.

Vernon Smith who edits the Washington, DC, CATS newsletter puts out the meatiest and most professional newsletter. It won second place in the TDM contest and was the top newsletter produced on a Sinclair computer.

This editor is pleased that some of what we publish is of interest to others. The Chicago area Nite-Times News copied Mike Mitchell's article and listing, "Telecommunications on the QL".

What was most interesting is that they got it from the Phoenix, AZ, based QZX which specializes in Sinclair Telecommunications.

We don't even exchange with them, yet! They got it from the group in Dallas. By the time it got back to us, a POKE had been added and a line changed. Congratulations, Mike!

We have also uncovered in the BCS archives exchange newsletters from 1983/1984, the heyday of Sinclair computing.

We really need a Coordinator of Newsletter Exchange to summarize the best for inclusion in this newsletter. It has gotten beyond the abilities of the editor to keep up. Volunteers, PLEASE!

- Peter Hale

TS 2068 MACHINE CODE
A Book Review

Introduction to 2068 Machine Code by Dr. Lloyd Dreger & Advanced 2068 Machine Code by Dr. Lloyd Dreger / S.M.U.G.

[A note to readers: The differences between "assembly language" and "machine language" are subtle and unimportant in this review. For the purposes of this article, the two terms may be considered equivalent.]

There are many good books that teach BASIC programming for beginning and advanced T/S 2068 programmers. However, the 2068 programmer interested in assembly language programming is faced with a shortage of information. Although there is no shortage of books on Z-80 assembly language, none of them cover the T/S 2068 and its operating system. An exception is the T/S 2068 Technical Manual, but that book was never intended to be a tutorial.

Dr. Lloyd Dreger, of the Sinclair Milwaukee Users Group, saw this problem and decided to do something about it. The result is two books. The first, a 213 page manual called Introduction to 2068 Machine Code, covers nearly every aspect of machine language on the 2068, including a tutorial on the basic concepts behind machine language programming. It was written with the assumption that the reader knows BASIC, but not necessarily machine code. However, those who already know something of Z-80 machine code will benefit just as much as the beginning student. The manual goes to the heart of the 2068, explaining the BASIC interpreter, the 2068 memory map, video modes & screen addressing, the system variables, and sound.

Armed with a thorough understanding of the inner workings of the 2068 and the basic concepts

(Cont page 18)

MAGAZINE REVIEWS

Orphan computers are lonely and have communication problems. It is not enough to be loved.

If that orphan is a Sinclair, the other problem is magazine costs. (Sinclair users operate on tight budgets and UK subscriptions also take forever to arrive.)

Six vital magazines are published in North America-land and maybe you don't know about them yet.

The grand daddy and guru of them all is **Time Designs Magazine**. Its 4000+ subscribers might also call it The Journal of Record of North American Timex/Sinclair Computing

TDM has goodies for each Sinclair model and clone. It also has the best (excellent) graphics.

Regular contributers include such luminaries as Fred Nachbaur, Stan Lemke, Paul Bingham, Mike de Sosa and others.

Each issue always has columns and programs for all three classes of Sinclairs, but its other value is that it is where everyone advertises - hardware, software, etc.

The May/June (Vol 4,No 4 - $3.00) issue lists 55 user groups and 94 companies that support Sinclairs.

Three magazines support TS 2068's (and Spectrum).

The most unusual is **Byte Power** on cassette - no typing listings - and mostly in machine code. It's bi-monthly on a flexible schedule but the software is extemely well written and often unique.

Back issues and software reprints are available.

UP-DATE Computer Systems comes quarterly. It is one third listings and supports every 2068 disk controller in each 50 page issue. It has excellent reviews and a sense of humor. Now supports QL.

The **SyncWare Group** publishes two bi-monthlies for the Sinclair and both are internationally reknown.

SyncWare News has been in print since 1983. It's for ZX-80 thru TS-2068 computers.

Quantum Levels is just for QLs.

Both are 16 page bi-monthlies for anyone who explores a Sinclair's full potential. No one who enjoys hacking should be without whichever one applies. Beginners may sometimes feel overwhelmed, but there is something for everyone.

As much as anyone, Syncware Group has its ear to the ground for the latest scoop and poop in the T/S community. It has good overseas contacts. The worst that can be noted is the (temporary) lag from editing to production and dearth of graphics.

Finally, do not overlook **Computer Shopper**, a hardy habitue of computer flea markets. There is an orphan column and much classified activity directed to Sinclairs.

The Bottom Line: You get change out of a twenty dollar bill for a year of most magazines.

Byte Power, 1748 Meadowview Ave., Pickering, ONT, CANADA L1V 3G8. US$32.00 for 6 issues. ($2.00 for Catalogue/Demo tape, a must buy)

Computer Shopper, P.O. Box F, Titusville, FL 32781. $2.95 at computer stores; by subscription, monthly. (Various special deals)

SyncWare Group, 602 S. Mill St., Louisville, OH 44641. $18.95/yr, bi-monthly, for SyncWare News or Quantum Levels. (Sample $2.00)

Time Designs Magazine, 29722 Hult Road, Colton, OR 97017. $16.95/yr bi-monthly. (Also get May/June 88 User Group issue for $3.00)

UP-DATE, 1317 Stratford Avenue, Panama City, FL 32404. $15.00/yr, quarterly.

LARKEN RAMDISK for T/S 2068
A Review

I recently received an order from Larken Electronics with Ramdisk on a medium-sized (3x7) circuit card and a replacement EPROM chip for the LKDOS disk controller cartridge.

The chip is an updated version of the disk operating system so that LKDOS can support the ramdisk.

The Ramdisk board has a feed-thru connector that plugs into the rear expansion buss of the 2068, and 4 empty sockets on the backside of the card. These sockets hold the ramchips, which must be bought separately. At one edge of the ramdisk board is a battery clip to hold two "AAA" batteries. With batteries installed, the contents of the ramdisk are preserved when the 2068 is turned off or Ramdisk is unplugged.

The LKDOS Ramdisk accepts up to four 32K ram chips without modification for up to 128K of storage. The board uses 32 x 8K static ram chips (part no 43256L or 62256L). To install memory beyond 128K, four more empty sockets must be soldered 'piggyback' to the first four ram chips, but this is relatively simple for anyone who has used a soldering iron.

Once installed, the ram is 'formatted' with a single command from basic and ready for use.

The Ramdisk behaves exactly like an ordinary disk drive in LKDOS, only a lot faster. The same commands are used; only where ordinary drives are numbered Zero to Three, the Ramdisk is drive Four. PRINT #4: CAT "" instantly responds with a catalog of the files under a heading "Disk drive name: RAMDISK". I could even change the Ramdisk drive name with Larry Kenny's recently released disk editor utility.

AUTOSTART files can be saved to the Ramdisk. Turn the 2068 on while pressing ENTER and the "J" key simultaneously will boot the AUTOSTART file from ramdisk instead of the regular disk drive.

Screen images loaded from the Ramdisk consecutively appear so quickly that an animation effect is achieved. I found that being able to load, save, boot and reload files to ramdisk continually instead of disk saves a lot of time and hassle. When I'm done, I dump the contents to a floppy.

This illustrates the Ramdisk's intended use to speed up existing programs when files are loaded and saved frequently, as opposed to increasing your disk space. Also, a 256K ramdisk stores more than an ordinary single-sided, double-density floppy disk.

The only problem I have encountered so far is some interference between the Ramdisk board and the 'stock' ZSI/O RS-232 card and modem at 1200 baud. However, I do not know if this is a problem with the Ramdisk, software, the ZSI/O, or plain power drain. (As it is, I have 7 devices chained to the back of my 2068). Other than that, I have found the LKDOS Ramdisk to be one of the most versatile tools yet for the 2068. Any serious 2068 user would do well to consider purchasing one.

The LKDOS Ramdisk is available from Larken Electronics at RR #2, Navan, Ontario, Canada K4B 1H9 and most local T/S dealers. Ramchips for the Ramdisk are about $14.00 apiece at Jameco (415-592--8097 to order). Chip sockets are about 50¢ each at Radio Shack.

- Jim Rodlin

(Ramdisking capacity is often the cheapest way to increase speed, since the most annoying delays in using a computer are when accessing media. For the QL, memory expansion plus ramdisk software make microdrive cartridges bearable. - Ed)

QUILL FILESAVER

More than once, my QL has mangled a long Quill document through poor disk or microdrive I/O. Either I removed the cartridge before closing a file, or the QL tried to write part of the file to a non-existent track on the disk or the QL crashed. The result is the same - any attempt to reload the file into Quill results in a **File I/O Incomplete** error, disasterous if you forgot to periodically backup your work.

One such encounter forced me to take a deeper look at how Quill files are stored. I could do this with "BGET", which reads a single byte from a file. Most disk systems and toolkits have this command.

I used BGET to print out the numeric value of every character in my partially-corrupted Quill file to learn its structure and so recover at least part of it.

The most striking discovery was that Quill files use the Null character [CHR$(0)] in place of Carriage Returns. I also noticed that a good portion of the Quill file is formatting information, stored towards the end of the file. It is mostly strange control characters. All the text was present, but some formatting information was cut off during the incomplete SAVE.

I tried renaming the file with a _LIS extension and importing that, but Quill didn't like the NUL's and stopped importing the file at the first occurance of a NUL.

So I had to figure out a way to extract the 'text' portion of my document and re-write it as straight ASCII. BPUT came to the rescue and the result is the listing below. In a nutshell, the program reads a byte from the damaged file in channel #5. If it is a null, it's changed to a carriage return [CHR$(13)]; if a control character (less than SPACE [CHR$(32)] but not CR), it is changed to a space. Otherwise, it is a regular ASCII character and left as is.

The now ASCII character is written to the new file in channel#6. The character is also printed to the screen so you can watch as it goes through the file. Then the loop repeats for the next character until there are no more characters to read. What is left is a plain text file which can be imported into Quill. I was able to recover a 15 page document this way, though it took about ten minutes to do so.

Hopefully you will never need to use this utility, but if you do, here it is.

- Jim Rodlin

```
 10 REMark       "Filesaver"
 20 REMark Corrupted Quill file
 30 REMark    recovery utility
 40 REMark       by Jim Rodlin
 50 REMark    Requires Toolkit
 60 REMark commands BGET & BPUT
 80 REMark In the public domain
 90 REMark Attribution requested
100 FLP_EXT: REMark or TK2_EXT
    enable extended command set
110 CLS
120 OPEN#5, "mdv2_MYFILE_doc"
    : REMark file to save
130 OPEN_NEW#6, "mdv1_BBS_lis"
    : REMark output ASCII file
140 REPeat loop
150  BGET#5, a%: REMark get byte
     and put in a%
160  IF a% =0 THEN LET a% =13
     REMark convert null to CR
165  REMark replace all control
     chars with SPACE
170  IF a% <32 AND a% <>13 THEN
     LET a% =32
180  REMark print character to
     screen for visual feedback
190  IF a% =13 THEN PRINT
     : REMark Carriage Return
200  IF a% <>13 THEN PRINT
     CHR$(a%);
210  BPUT#6, a%: REMark write
     ASCII byte to _LIS file
220 END REPeat loop
```

BCS PROGRAMMING PROJECT

The BCS Sinclair/Timex User Group has launched a project to develop a state-of-the-art BBS program for the T/S 2068 computer. We are looking for volunteers to participate in the project. A knowledge of Pascal or Z80 Assembly language would be helpful.

We will build advanced features into the software, including 1200 baud, Xmodem, and networking with other 2068 BBSes. The software will be developed under LKDOS V3.0. You need not live in Massachusetts to participate.

If you are interested, phone Jim Rodlin at (508) 481-2155, or write to him at 66 Chandler St., Marlboro, MA 01752. By modem, he can be contacted on the BCS TIMEWARP BBS at (508) 481-0555, 8/1/N 300 baud.

BINGO

There are one or two Sinclair computer users who play Bingo, or whose family does. From the 'HACKER', the official newsletter of the Timex/Sinclair User Group of Las Vegas, NV, comes this gem for the 2068 that prints out BINGO cards on a 2040 printer.

QL users may want to convert to a SuperBASIC program to get a better understanding of AT, BEEP, BORDER, FOR and TAB (use TO instead), which have slightly different constructs.

LPRINT, of course, is unavailable in SuperBASIC. Instead, insert line 5 : 'OPEN#3,ser1' to open a channel to the printer, then use 'PRINT#3,' instead of 'LPRINT'.

This is also an easy program to change GO SUB routines to SuperBASIC procedures.

```
 1 REM "BINGO"
 2 BORDER 1: PAPER 6: INK 0
 3 CLS : CLS
 4 GO TO 600
 6 PRINT AT 2,1;"Welcome to the
   Bingo Parlor";AT 3,4;"Here ar
   e your options:";AT 6,4;" 1
   - PLAY BINGO"; AT 8,4;" 2  -
   PRINT OUT BINGO CARDS";AT 10,
   4; " 3  - MASTER CALL SHEET";
   AT 12,4;" 4  - STOP"; AT 20,1
   ;"Enter Option Number (1-4):"
 8 INPUT z: CLS
10 GO TO 20*(z=1)+90*(z=3)+700*(
   z=3)+85*(z=4)
20 RANDOMIZE
25 DIM y$(75)
30 LET l$="BINGO"
35 BORDER 3: PAPER 7: CLS : CLS
   : PRINT AT 2,10;"Bingo Caller
   ";AT 8,6;" CALLING: ";AT 12,6
   ;"Press ENTER for next call";
   AT 16,3;" N  for New Game";AT
    18,3;" Q  to Quit"
40 FOR z=1 TO 75
50 LET r=INT (RND*75)+1
55 IF y$(r)="*" THEN GO TO 50
60 PAUSE 80: PRINT AT 8,17;" ";l
   $(INT (r/15+.95));"-";r;: BEE
   P .2,4
65 LET y$(r)="*"
70 IF INKEY$="" THEN GO TO 70
72 IF INKEY$="N" OR INKEY$=" n"
   THEN CLS: GO TO 6
74 IF INKEY$="Q" OR INKEY$=" q"
   THEN PAUSE 200: CLS: STOP
76 PRINT AT 8,18;"          "
80 NEXT z
85 CLS : STOP
90 PRINT AT 11,1;"HOW MANY CARDS
    DO YOU WANT?";
95 INPUT cards: PRINT AT 11,30;c
   ards: PAUSE 80
100 FOR x=1 to cards
101 LPRINT ,,,,
102 LPRINT "-------------------
    ----------"
103 LPRINT ,,,,
105 LPRINT : LPRINT "      THE HAM
    MER HUTCH CASINO": LPRINT : L
    PRINT "   * B  * I  * N  * G
     * O  *"
110 DIM y$(75)
115 DIM a(5)
120 GO SUB 300
125 FOR x=1 TO 5
130 GO SUB 400
135 FOR z= 1 TO 5
140 LET a(z)=INT (RND*15+(z-1)*15
    +1)
```

```
145 IF y$(a(z))="*" THEN GO TO 14
    0
150 LET y$(a(z))="*"
155 NEXT z
160 IF y=3 THEN LET a(3)=0
165 LPRINT TAB 3;"*";TAB 5;a(1);T
    AB 8;"*";TAB 10;a(2);TAB 13;"
    *";
170 IF y=3 THEN LPRINT TAB 14;"FR
    EE";
175 IF y<>3 THEN LPRINT TAB 15;a(
    3);
180 LPRINT TAB 18;"*";TAB 20;a(4)
    ;TAB 23;"*";TAB 25;a(2);TAB 2
    8;"*";
185 GO SUB 400
190 GO SUB 300
195 NEXT y
200 FOR z=1 TO 5
205 LPRINT,,
210 NEXT z
215 NEXT x
220 CLS
225 GO TO 500
300 LPRINT "    ******************
    ********"
305 RETURN
400 LPRINT "    *     *     *     *
      *     *"
405 RETURN
500 PRINT AT 4,4;"Choose your nex
    t Option:";AT 7,8;"  1  - Mor
    e Cards";AT 9,8;" 2  - Play B
    INGO";AT 11,8;" 3  - Master C
    alling Sheet";AT 13,8;" 4  -
    STOP";AT 20,1;"Enter Option N
    umber (1-4):"
505 INPUT q$: PRINT AT 20,28;q$
510 IF q$="1" THEN CLS: GO TO 90
515 IF q$="2" THEN PAUSE 80: CLS:
     GO TO 20
520 IF q$="3" THEN CLS: GO TO 700
525 IF q$="4" THEN CLS: STOP
600 PRINT AT 4,8;"WELCOME TO THE"
605 FLASH 1: PRINT AT 6,8;" HAMME
    R HUTCH ";AT 7,8;"     CASINO
       ": FLASH 0
610 PRINT AT 12,0;"COME IN AND SI
    T IN ONE OF OUR   COMFORTABLE
     CHAIRS IN OUR BINGO PARLOR.

      PRINT OUT A COUPLE OF CARDS,
          GRAB A PEN OR PENCIL, AN
    D HAVE SOME FUN."
620 PAUSE 300: CLS: GO TO 6
700 BORDER 0: PAPER 7: INK 0
705 CLS: CLS
706 PRINT AT 10,1;"Master Call Sh
    eet now printing"
708 PRINT ,,,,,,
710 LPRINT "Master Call Sheet"
715 LPRINT "-------------------
    ----------"
720 LPRINT "B"
722 LPRINT "1  2  3  4  5  6  7
    8"
724 LPRINT "9 10 11 12 13 14 15"
726 LPRINT "I"
728 LPRINT "16 17 18 19 20 21 22
    23"
730 LPRINT "24 25 26 27 28 29 30"
732 LPRINT "N"
734 LPRINT "31 32 33 34 35 36 37
    38"
736 LPRINT "39 40 41 42 43 44 45"
738 LPRINT "G"
740 LPRINT "46 47 48 49 50 51 52
    53"
742 LPRINT "54 55 56 57 58 59 60"
744 LPRINT "O"
746 LPRINT "61 62 63 64 65 66 67
    68"
748 LPRINT "69 70 71 72 73 74 75"
750 LPRINT "-------------------
    ----------"
755 LPRINT ,,,,,,,
800 BORDER 0: PAPER 7: INK 0
810 CLS: CLS
820 GO TO 6
900 SAVE "BINGO" LINE 1: BEEP .75
    ,12: BEEP .47,8
999 STOP
```

EAST COAST SINC FEST

The Capital Area Timex Sinclair Croup (CATS), Washington, DC has announced it will sponsor a Sinclair Computer Fest in May, 1989.

Major Vendors are participating and it is expected to be the outstanding Sinclair Fest.

A major advantage of this fest is its central location on the East Coast and will expect attendance from some of the nearby centers of T/S mania.

The Fest will be at the Howard Johnsons in New Carrollton, MD, close by the METRO with connections to the entire DC area.

The BCS will sponsor a table and arrange for group transport and accomdations. Watch this space for further details.

ABOUT TOOLKIT II

Are there any QL owners who if they don't have TK2, have not been strongly advised to get it?

Anyone who has keystroked COPY mdv2_fred to mdv1_fred a zillion times to backup media finds WCOPY worth the price alone.

Certainly, the Toolkit II manual is a forbidding document. There is no tutorial; just a list of keyword extensions with limited and sometimes incomplete directions on their use.

There is not even a clue as to where to get your feet wet, so a sampling of easy and immediately useful keywords is offered here.

Toolkit II extensions that are easy and that you will enjoy are: ALARM, CLOCK, DLIST, EX, EXTRAS, EW, FREE_MEM, RENAME, SPL, SPLF, STAT, TK2_EXT, VIEW, WCOPY, WDIR, WMON, WREN, WTV, and WSTAT.

All can be explored from direct mode and you can't get in (much) trouble (except for difficulty understanding the manual).

But a lot of people are missing out on three excellent features incorporated in Toolkit II: ED, ALTKEY and last line recall.

Last line recall <ALT><ENTER> is usually useful in direct mode in SuperBASIC. It recalls all characters since the last ENTER. Done several times in sequence, it recalls several lines at a time. Once recalled, lines can be edited or reexecuted. It also works from Psion programs, as a quick and dirty copy, up to Psion's 22 character buffer limit.

ALTKEY is worth exploring. It is buried in the back of the manual in Section 21 and little is said about it. Essentially it permits writing macros. The syntax is **ALTKEY** 'x','....' where x is any one of the QL's character set. In the example, x is lower case. Everything after 'x', and within the quotes happens when the combination <ALT><x> is pressed. If <ENTER> is desired for executing, the ALTKEY definition can end with ,'' (comma and two single quotes), which is <ENTER>.

How to use ALTKEY is not conceptually easy. One use is ALTKEY '1', CHR$(240)&'lflp1_setup',''. This line is incorporated in my Quill boot. The same disk also has the file 'setup_doc' on it.

Setup_doc is my style sheet for correspondence. It has my return address, the date and salutation position and no footer. I don't reorganize Quill's default page each time I write a letter; I press <ALT><1>.

Programmers assign ALTKEYs to oft used keywords (without an ENTER) to emulate single-stroke keywords popular in earlier Sinclair's. A boot with a batch of user defined ALTKEYs and a final line NEW is run at the start of programming. NEW clears SuperBASIC memory (but not the ALTKEY definitions.)

Finally, there is ED that invokes a full screen SuperBASIC editor. The cursor keys move the cursor over the full screen. They can be combined with ALT and SHIFT for special effects. It is easier to edit existing programs with Overwriting (SHIFT F4, as in Quill).

SuperBASIC programs often have a number of duplicate or near duplicate lines. Editing a line number replicates the line at a new position and retains the old line. If slight modification is required, overwriting speeds any needed changes.

For fun, slip in ED as a line anywhere in a SuperBASIC program. Run it. Predictably, when the program reaches ED, it halts and goes into full screen edit mode. Now press <ESC>. What happens?

-Peter Hale

COLUMNIFYING QUILL

by Mike Padlipsky & Peter Hale

The trouble with having been a working programmer in one's youth is that you spend the rest of your life under the constant threat of getting readdicted to that particular form of obsessive-compulsive behavior.

(The trouble with being an ex-programmer who's also a Published Author, by the way, is that you feel obligated to spend much more time describing programs than writing them. This can boggle a newsletter, unfortunately. So between my rewriting and his compressing, what you're seeing is about 60% of what your Editor originally got. [In case anybody cares, The Book is **The Elements of Networking Style**, Prentice-Hall, 1985.])

So when, for obscure reasons we needn't bother with here, I decided the other week that I needed to be able to print a Quill document in Double Strike Condensed mode, in columns, I knew I had a problem. That meant I'd have to write a program to do it. And that meant I'd have to run the risk of getting rehooked, just as I had for several months last year, shortly after acquiring my QL. As a matter of self-protection, then, I consciously did the program in the simplest, laziest way I could think of, so as to re-expose myself for only an hour or two, and sat back to enjoy it.

There's even trouble with sitting back and enjoying your columnifying program, though. Part of the enjoyment was bragging how nifty it was to Peter Hale, who'd been helpful in the planning stages and who'd expressed a desire for such a program to produce this very newsletter. Zap. Temptation again. He wanted it to be more "user friendly"; suggested all sorts of embellishments, indeed.

Having already produced my "Sweetened Condensed" columnized document (it's lovely, thanks), I managed to hold firm, though, and convinced him--in part by promising to do the writeup myself: more trouble--to do his own version of the thing. So what follows is actually a collaborative effort. Peter assures me I'm entitled to take credit for the semi-elegant SuperBASIC gimmick, so I will; but anything you don't like about the user interface, blame on him.

All I ever wanted was to run a 65 character wide Quill document off into a file named colify_lis, having forced it to have an even number of whole pages by blanking enough lines at the bottom to come out right, make it print out in columns, and just run off the end because I remembered from the Lost Months of a year ago that there was something funny about end-of-file detection with Quill.

USAGE

The first thing to do is to make a Quill document with an appropriately narrow line. The table at the end of the article gives maximum line lengths for anticipated type faces. In my short version (here for those who don't want all the typing for Peter's version) choose a non-0 margin. Its width will also be the width of the column separator.

How to end the file is important in Peter's version. The last line has to consist of only a <CTRL-SHIFT-K> and an Enter. This makes life easier than coping with the imponderables of how Quill ends _lis files. For my version just pad the Quill file out with one or two Page Breaks, to end up with an even number of pages when you're done. Close the channels by hand when it runs off the end of the file.

Now run the Quill document off to a _lis file. I.E. type colify over the "printer" prompt as the last step of the Print command;

you don't even need to specify the _lis part, it's furnished by the Quill program automagically. (The version Peter will stick in lets you select any file name allowed by Quill.)

The next trick is to invoke the columnifier program by lrunning whatever name you saved it under. My version just gives two columns of Sweetened Condensed, provided the Quill document is no more than 66 columns wide. Fire up Peter's and it'll start asking you about various things like linewidth, typeface and so on. It will even let you alter the left margin and respecify the width of the column separator.

Most of the usage is as nearly self-explanatory as these things ever are. If things really seem fouled up, be of good cheer: you can always get out by hitting <CTRL><SPACE>, since it is, after all, a SuperBASIC program.

The one obscure point, in my humble but dogmatic opinion, is this business of "text frame". (That's the measured width of two columns of text, <u>including the column separator</u>, for a chosen type face without regard for left margin adjustments. The 'out of range' prompt takes into account margin adjustments.)

Not everyone will use the maximum width for a given type face. To avoid a cramped look, it may be necessary to widen the column separator. If the output is to be reduced, a full 8 inches of text frame may be excessive.

There are doubtless a number of fine points I'm overlooking here, but with any luck at all, that's all you need to know about columnifying prior to using the program - except for the Limitations mentioned below, anyway. If it isn't, the REMarks in the program listing might well help even if you're not a programmer and would otherwise be totally disinclined to look at them.

Another point about my lazyman's version might be in order: if you decide to use it rather than the full version and want to adapt it for other than Condensed, Peter's code should help in deciphering what you have to send to the printer. The thing to be wary of is that you <u>must</u> diddle the typeface before you fiddle the tab setting. (It might seem obvious, but overlooking it doubled the time I spent writing the code.)

Oops. One more feature: Peter's version has the option to run the output into a file (which will have same "first name" as your _lis file, with the suffix _col) if you want to copy a number of _col files to serl all at once.

LIMITATIONS

<u>First and foremost</u>: if the Preamble in your printer driver does a reset (i.e., has ESC,@), get rid of it!!! It took me hours of headscratching and a couple of lengthy conversations with Peter to figure out why the full-up program didn't let my machine make a draft of this very article. Sigh.

The lack of page numbering is noted and won't be apologized for; my psychic well being is at stake here. (Perhaps Peter will be shamed into doing it himself; check for notes after the listings.) [Why wait? You know that Editors have no shame! -Ed]

The main thing to realize is that this is a fairly dumb little program at heart. All it does is read 66 lines out of a _lis file, change the CRs at the end of each line to Tabs, then read the next 66 lines and concatenate them with the first 66 (one at a time, of course) before doing a SuperBASIC PRINT statement with them to the printer.

So if you're doing a lot of boldfacing or underlining in your document, be aware that if you don't turn them off at the end of each line, they'll slop over into

either the righthand column or the next lefthand line when printing. Similar considerations apply to meta-Quill fancies, like selective italicizing by having appropriate Translate entries in your printer driver.

Sorry about that, but it really would have become unacceptably hairy to have the program keep track of that sort of thing for you. If you don't have a feel for the problem being described, don't worry; if and when you get funny looking output, it'll be clear how to deal with it whether or not the semi-explanation made sense just now.

There are also some limitations in the printers you're likely to hook up to a QL, like the bizarre fact that Boldface temporarily disables Condensed, or the subtle point that if you do run the output into a file you can't reimport the file into Quill for further fiddling in columnized form because one of the printer control codes ends in <NUL>, which makes Quill stop importing. But I won't try to remember any others than those two since there isn't anything we can do about them and I've probably gone on too long anyway.

THE LISTINGS

Here comes mine, in all its minimalist glory:

```
110 DIM lcol$(66,80)
120 DIM rcol$(66,80)
130 OPEN_IN #3,'flp2_colify_lis'
140 OPEN #4,'ser1'
150 PRINT #4,CHR$(27);'@';
    CHR$(27);'G';CHR$(15);
    CHR$(27);'D';'B';CHR$(0);
160 FOR i=1 TO 66
170  INPUT #3, lcol$(i)
175  j=CHR$(13) INSTR lcol$(i)
180  lcol$(i,j)=CHR$(9)
190 END FOR i
200 FOR i=1 TO 66: INPUT #3,
    rcol$(i)
210 FOR i=1 TO 66: PRINT #4,
    lcol$(i) & rcol$(i)
220 GO TO 160
```

If the 'B' in Line 150 puzzles you, it's just the lazyman's way of spelling chr$(66). And for any programmers watching, I do have Dykstra's permission for 220.

More than enough of this foolishness. Over to Peter:

There really isn't much to add to Mike's prose. The three pages of listings are tedious to type in. If you do, save it as Bi_col_bas by starting with line 2000, a wee procedure to help you save the program by just typing s <ENTER>.

To get the listing and somewhat different instructions from those above, send a microcartridge and $3.00 handling to the Sinclair group at the BCS.

Remember: No header and no footer in the Quill document and no Reset in the printer_dat Preamble for when you create the _lis file. Terminate any enhancement where the line ends and reset it where the line begins below.

In the long version set the left margin to zero; Bi_col_bas lets you adjust margins and column separator widths later. And don't forget how many columns you set the right margin for. Let the last two characters in the name of the _lis file be the number as a memory aid.

Here are the maximum number of columns per line:

Type face	Maximum characters/line
Double-wide	18
Standard (Pica)	38
Elite	46
Condensed	66

If you anticipate proportional typeface, the limits can be increased by a factor of 1.1.

There's nothing in Bi_col_bas that shouldn't be feasible in another BASIC to Bi-columnify any ASCII-fied document and it should be most elegant on a T/S 2068.

```
1 REMark       BI-COLUMNIFIER
2 REMark     Prints a _lis file
3 REMark as two columns to a page
5 REMark     by  Mike Padlipsky
7 REMark         & Peter Hale
8 REMark Sinclair/Timex User Group
9 REMark  Boston Computer Society
10 REMark     One Center Plaza
11 REMark     Boston, MA 02108
13 REMark  Attribution requested
14 :
15 REMark Progam assumes Epson compatible printer codes
16 REMark See line 1500 & ff for codes needing changes
17 :
100 init: open_file: char_col_line: CHAR_wid: CHAR_2: verify
110 print_file: PRINT_col
120 CSIZE 3,0: AT 20,5: INK 0: PRINT "; FINIS !"
130 BEEP 0,50: PAUSE (150): BEEP: STOP
140 :
150 DEFine PROCedure gky
160  an$=INKEY$(-1): an=CODE(an$): PRINT!an$
170 END DEFine gky
180 :
200 DEFine PROCedure init
210  WINDOW 512,256,0,0: INK 6: PAPER 2: CLS: CSIZE 3,1
220  WINDOW 312,220,15,0: BORDER 2,6: CLS: AT 4,2
230  PRINT "BI-COLUMNIFIER": CSIZE 1,0
240  PRINT\"    by Mike Padlipsky & Peter Hale"
250  PRINT\\\,"Press <SPACE> to run": PAUSE: CLS
260  k=0: mar=0: tb=4: r=1: n$=CHR$(0): p$=n$: t$=n$: w$=n$
270  c_1=5: typ$="Standard ": nlq$="Draft": spc$=typ$
280 END DEFine init
290 :
300 DEFine PROCedure char_col_line
310  PRINT\"  Enter number of characters per"
320  INPUT "  line in original text:"!c_c_1$ : c_c_1=c_c_1$
330  IF c_c_1>66
340   PRINT\" "!c_c_1!"characters too wide to"
350   PRINT "  bi-columize the document."
360   PRINT "  66 is maximum for condensed type."
370   PRINT\"  Re-do your document.": STOP
380  END IF
390 END DEFine char_col_line
400 :
500 DEFine PROCedure CHAR_wid
510  PRINT\"  Select character width": PRINT
520  IF c_c_1<19 THEN PRINT,"<D> ouble wide"
530  IF c_c_1<39 THEN PRINT,"<S> tandard (Pica)"
540  IF c_c_1<47 THEN PRINT,"<E> lite"
550  IF c_c_1<67 THEN PRINT,"<C> ondensed"
560  PRINT\"  Enter choice (d/s/e/c)";: gky
570  SELect ON an
580   =68,100: w$=CHR$(27)&"W"&CHR$(1): wid$="Double Wide"
590   =80,112,83,115: wid$="Pica": c_1=10
600   =69,101: w$=CHR$(27)&"M": wid$="Elite": c_1=12
610   =67,99: w$=CHR$(15): wid$="Condensed": c_1=17
620   =REMAINDER : CLS: RUN
630  END SELect
640 END DEFine CHAR_wid
650 :
```

```
700 DEFine PROCedure CHAR_2
710  PRINT\"  Italic character type? (y/n)";: gky
720  IF an$=="y" THEN t$=CHR$(27)&"4": typ$="Italic"
730  IF c_i<12
740   PRINT\"  Proportional spacing? (y/n)";: gky
750   IF an$=="y"
760    p$=CHR$(27)&"p"&CHR$(1): spc$="Propor'l ": r=1.1
770   END IF : END IF
780  spc$=spc$&(c_i*r)&" char/inch"
790  PRINT\"  N.L.Q. Printing mode? (y/n)";: gky
800  IF an$=="y" THEN
810   REMark c_i<12 if your Epson can't do NLQ in elite
820   IF c_i<17
830    n$=CHR$(27)&"x"&CHR$(1): nlq$="Near Letter Quality"
840   ELSE
850    n$=CHR$(27)&"G": nlq$="Double Strike"
860   END IF : END IF : CLS
870  c_l=c_c_l/r: tab=INT(tb*r+c_l): d_f=INT(10*(tab+c_l)/c_i+.5)
880 END DEFine CHAR_2
890 :
900 DEFine PROCedure verify
910  PRINT,"  DOCUMENT FORMAT"
920  PRINT\"  Char fount",typ$
930  PRINT "  Char size",wid$
940  PRINT "  Spacing",spc$
950  PRINT "  Text frame",d_f/10!"inches"
960  PRINT "  Printing mode",nlq$
970  PRINT\"    Left   Char/ Column   Rel tab Text"
980  PRINT "  Margin line   Sep'tr position frame"
990  PRINT\"     ← →              ↑ ↓"
1000  PRINT\TO 7;"Press cursors to adjust"\TO 11;"<ENTER>  to set"
1010  REPeat loop
1030   AT 10,4: PRINT mar,!!c_c_l,!tb,!tab!!!!!d_f/10;"'"
1040   PAUSE: an=KEYROW(1)
1050   SELect ON an
1060    =2: mar=mar-1
1070    =16: mar=mar+1
1080    =4: tb=tb+1
1090    =128: tb=tb-1
1100    =1: EXIT loop
1110    =REMAINDER
1120   END SELect
1130   tab=INT(tb*r+c_l): d_f=INT(10*(tab+c_l)/c_i+.5)
1140   AT 12,6: PRINT FILL$(" ",25)
1150   IF mar<0 OR (mar*10/c_i+d_f)/10>8 OR tb<3
1160    AT 12,6: INK 0: PAPER 6
1170    PRINT " Out of range; try again ": INK 6: PAPER 2
1180   END IF
1190  END REPeat loop
1200  AT 16,7: PRINT "Is this correct?  (y/n)";: gky
1210  IF an$=="y" THEN RETurn verify
1220  PRINT "  Then you must rework your Quill doc"
1230  PRINT\"  Exit to rework Quill doc? (y/n)";: gky
1240  IF an$=="y" THEN CLS: STOP
1250  PRINT " Try another document format? (y/n)";: gky: CLS
1260  IF an$=="y" THEN GO TO 100: ELSE : STOP
1270 END DEFine verify
1280 :
1300 DEFine PROCedure open_file
1310  OPEN #5,con: WINDOW #5,120,256,330,0: BORDER #5,2,4
```

```
1320  INK #5,6: PAPER #5,0: CLS #5
1330  INPUT #5,\"   Document on"\\"   DRIVE?"!dv$
1340  IF LEN(dv$)=1 THEN dv$="flp"&dv$
1350  REMark Number of fields = 66 lines/page
1360  REMark Field length = 99 columns
1370  DIM lcol$(66,99): DIM rcol$(66,99)
1380  IF dv$(LEN (dv$))<>"_" THEN dv$=dv$&"_"
1390  DIR #5,dv$&"_lis": INPUT #5,\" Select _Lis"\\" FILE:"!file$
1400  REMark Line 1390 uses Toolkit II DIR ability display only files
1410  REMark that end in "_lis".  You may have to edit out &"_lis"
1420  IF NOT "_" INSTR file$ THEN file$=file$&"_lis"
1430  LET temp$=dv$&file$(1 TO ( "_" INSTR file$))&"col"
1440 END DEFine open_file
1450 :
1500 DEFine PROCedure PRINT_col
1510  REMark Printer control codes are for Epson printer
1520  REMark You may have to change for other printers
1530  PRINT #4,CHR$(27)&"@";: REMark printer reset
1540  IF n$<>CHR$(0) THEN PRINT #4,n$;: REMark See Lines 830/850
1550  IF p$<>CHR$(0) THEN PRINT #4,p$;: REMark See Line 760
1560  IF t$<>CHR$(0) THEN PRINT #4,t$;: REMark See Line 720
1570  IF w$<>CHR$(0) THEN PRINT #4,w$;: REMark See Lines 580 to 610
1580  REMark Following line sets left margin, then horizontal tab
1590  REMark "IF an$<>'f'" and 1610 to reimport file to Quill
1600  PRINT #4,CHR$(27)&"l"&CHR$(mar)&CHR$(27)&"D"&CHR$(tab)&CHR$(0);
1610  REMark END IF
1620  REPeat col_PRINT
1630   FOR i=1 TO 66
1640    INPUT #3,lcol$(i)
1650    j=CHR$(13) INSTR lcol$(i)
1660    lcol$(i,j)=CHR$(9)
1670    REMark Œ=CTRL SHIFT K
1680    IF "Œ" INSTR lcol$(i) THEN k=i: EXIT i
1690   END FOR i
1700   IF k: FOR i=k TO 66: lcol$(i)=CHR$(9)
1710   IF NOT k
1720    FOR i=1 TO 66
1730     INPUT #3,rcol$(i)
1740    IF "Œ" INSTR rcol$(i) THEN k=i: EXIT i
1750    END FOR i
1760   END IF
1770   IF k: FOR i=k TO 66: rcol$(i)=CHR$(13)
1780   FOR i=1 TO 66: PRINT #4,lcol$(i);rcol$(i)
1790   IF k THEN EXIT col_PRINT
1800  END REPeat col_PRINT
1810  CLOSE #3: PRINT #4,CHR$(27)&"@";: CLOSE #4: CLOSE #5
1820 END DEFine PRINT_col
1830 :
1900 DEFine PROCedure print_file
1910  PRINT\"  Print to Paper or File (p/f)";: gky
1920  IF an$=="p" THEN OPEN #4,ser1: ELSE an$="f"
1930  OPEN_IN #3,dv$&file$: PRINT #5,\dv$&file$\"  will be printed"
1940  IF an$='f' THEN PRINT #5,"  as:"!temp$: OPEN_NEW #4,temp$
1950  PRINT #5,\\"     PATIENCE!"
1960 END DEFine print_file
1970:
2000 DEFine PROCedure s
2010  DELETE flp2_B1_col_bas: SAVE flp2_B1_col_bas
2020 END PROCedure s
```

behind machine code thinking, the reader dives right into machine language programming with a look at the CPU and how it works. The Z-80 assembly instruction set is examined in detail, with examples demonstrating their use. Later, the author touches on advanced concepts like bank switching, the floating point processor, the function dispatcher, and how peripherals (such as printers, disk drives, and modems) work. Six appendices provide helpful tables and information for the reader.

One complaint I have is with a PRINT routine in appendix B that had bugs in it. The author uses the program to teach basic techniques to his machine code classes at SMUG, so a detailed explanation of the program and what's going on is not included. I suspect that the bugs are deliberate and meant for his students to correct. However, there is no mention of this in the manual. It can be irritating for the beginning reader to be faced with a buggy program with no hints as to how to correct it, especially when he doesn't expect any bugs in the first place.

Book two, Advanced 2068 Machine Code Volume 1, covers graphics and advanced screen programming. If you want to write that ultimate "Paint" program, or you are interested in the 2068's advanced video modes, this book is what you are looking for. It begins with a few notes about assemblers and compilers for the 2068. The second chapter starts you out with a few simple routines that should be in every programmer's toolbox (screen clear, locate screen address, etc). Chapter three includes an article (originally from the SMUG Bytes newsletter) about bank switching BASIC programs with the Aerco disk drive's expanded memory banks. The author expands upon it with a discussion about using advanced screen modes while bank switching.

The next three chapters are devoted to the TS 2068's advanced video modes: 64 and 80 (yes 80) column screens, dual screen mode, and high resolution graphics mode. The author provides many examples and a generous amount of source code, including complete, debugged versions of the advanced video mode routines from the T/S 2068 Technical Manual.

The remainder of the book is about game programs and the ideas behind them. 'Sprites' are examined and a super sprite driver is developed that puts the Tech Manual routines to shame. It culminates in a shoot-em-up arcade game that involves some complicated timing, interrupts, and even drawing on the border. Finally, the book examines fractals and how to develop the scene portion of graphical adventure games.

For those who aspire to become crack Z-80 assembly language wizards on the 2068, these two books are a must. If you are a beginning machine language programmer, you will not find a more informative and well-thought out tutorial anywhere.

Dr. Dreger is currently working on Advanced 2068 Machine Code Volume 2, which will cover BEEP, sound, and music on the 2068, and a closer look at the T/S 2068 ROM routines, to be released in 1989.

A companion to these books, The Timex/Sinclair 2068 ROM Manuscript, also by Dr. Dreger, is a disassembly of the 2068 ROM with comments and cross-references to the Sinclair Spectrum ROM. It is a valuable reference for assembly language programmers of all leagues.

The books are distributed by S.M.U.G., the Sinclair Milwaukee Users Group, at Box 101, Butler, WI 53007. They are also carried by most Timex/Sinclair dealers. Order yours today.

- Jim Rodlin